2005 VALUATION ACTUARY SYMPOSIUM*

Orlando, Fla.
September 22–23, 2005

Session 8TS
Liability Modeling Concepts

Instructors: JEFFERY A. FITCH
URI N. SOBEL
ROBERT W. WELANDER

Summary: Model building is required for much of the work completed by actuaries including cash-flow testing, actuarial appraisals, C-3 Phase II RBC, VACARVM and SOP 03-1. This session provides an introduction to model building for those who have limited experience or need a refresher. Panelists discuss the basics of modeling including: model point selection; appropriate model granularity; static and dynamic model validation; efficient model refresh processes; assumption development; and reconciliation to actual financials. At the conclusion, attendees have a better understanding of the model building process.

MR. URI N. SOBEL: This is going to be somewhat of a beginner's guide to liability modeling concepts. I'm a consultant at Milliman in New York. Robert Welander is one of my co-presenters. He's going to be talking about efficient model development process and assumption development. Jeff Fitch is going to be covering model organization and validation. I'm going to be covering model point selection and appropriate model granularity. The topics I hope to cover are the considerations in deciding whether to use seriatim or map-type models, considerations in how to get and understand the data you're going to be modeling, some considerations in performing the mapping and a couple of miscellaneous observations and considerations.

So, the first decision you're going to have to make in your liability modeling is whether to do any mapping at all or simply to run every policy through your model seriatim. Most often this is just going to end up being somewhere in the middle of that spectrum, but the first set of decisions you're going to make is going to have

Note: The chart(s) referred to in the text can be found at http://handouts.soa.org/conted/cearchive/valact05/2005valact_handouts.htm.

to do with where in that spectrum you want to lie. One reason you may choose to run on a seriatim basis is simply that in the end, if you end up with any sort of validation issues or projections that are producing what looked to you like unreasonable answers or anything seriatim, you'll at least know what is not causing those problems, and it's one fewer thing to have to examine.

Also, importantly, you should consider what you're going to be using the results of your model for. If you're going to be developing a fairly precise number that needs to be correct to the dollar, you may want to go more on the granular side. On the other hand, if you're simply doing sensitivity testing, and you need to know magnitude and direction of change, perhaps a less granular model will do. For valuation work and pricing there are other considerations. In some pricing cases there's no such thing as seriatim. There's no defined, fixed universe of all policies that you could be modeling.

You should consider what the resources available to you are, particularly your computer resources. Do you have a grid or set of computers that you can dole out all your jobs to? One word of caution on this issue, however, is just because you have 3,000 machines available to you in your office doesn't necessarily mean that that's going to help you if you only have a couple of the really fast ones there. You don't want to come in the next morning and look at your results of your stochastic projections and find out that three of the slow machines still have some jobs, and you're not going to be done for a couple of hours. But if you have a large grid of computers, that may mean you could use a more granular model, and runtime wouldn't be as much of an issue.

Availability of data is a consideration that is important to keep in mind. We had a project recently where we had a very granular model, and, in fact, we ended up spending a great amount of time searching for good policy value, premiums and cash values for a whole lot of model cells. That time was probably better spent elsewhere, and so we probably would have been better off with a less granular model for the purposes we were using it for, and we wouldn't have had those sorts of issues.

Often you'll find the same business is being modeled in more than one context. You may have a short-term GAAP earnings model. There might be a strategic planning model out there, some cash-flow testing and some embedded value. What often happens is results from these various models of the same business end up going up the chain somewhere, and somebody sees results of the same business being modeled by two different models, and you end up having to explain why the results are different. So, if you can use the same model between these two contexts, that could help you ease some of that pain, although there is a tradeoff there. One alternative is to perhaps start at the most granular level with a lot of cells for your most precise applications and then try to crunch down as needed, crunching down on perhaps issue month, issue age and things like that.

Well, whether you decide to do things on a map basis or on a seriatim basis, you're going to need the data, and unlike the usual proverb less is more, I believe in this case more is more. It's better off to get the raw actual data with as many fields as you can think of and getting the data seriatim. If not seriatim, it’s best to get it minimally grouped. Once we get the data enhanced, typically we’ll throw it into a database program. We'll use Access typically, and we will produce some simple marginal distributions. We'll look at where the business is in terms of different plans and different issue ages. We’ll look at what the distribution of business by sex, underwriting status, smoker, issue year and perhaps issue month are. Depending on the type of business we would also produce other marginal distributions. If sales channel will be important, we'll try to take a look at that, or we may look at funding level for some flexible premium products, in-the-moneyness for certain types of secondary benefits and so forth.

Once we have a sense of where the business lies in that regard we'll then produce some simple joint distributions. So, for each plan in our actual inforce we'll go and try to look at the distribution of our business, let's say, by sex and issue age and smoker. Sobel page 2, Slide 4 shows very simple example, not a real one, where we took the inforce data that we got from our client and took a look at the distribution by issue age and tried to see how important it is to include each of these as separate cells. You look at the 0- to 29-year-olds, and you may decide, well, that's too small a block. It's not that important to model them separately. Maybe I'd lump them in with my 30- to 39-year-olds. You would have a similar analysis on the 60-plus-year-olds. In terms of counting units you'd think that's de minimis. On the other hand, you look at those reserves and you look at the premiums they're paying, and it's significant enough to include them as separate cells.

A lot of the mapping that you're going to be doing is going to involve decisions as to how much to combine on the characteristics from the marginal distributions that we put together—plan, issue age, sex. Typically you distinguish your model plans by different maturity periods and different maturity ages. If it's a limited premium product, you may have different plans for the different premium paying periods. Annuities would have different model plans for different annuitization periods and age, and so on. Even if the business is very similar in the two model plans, model plans sometimes may be distinguished if the business is being backed by separate asset segments or if for some company-context-related reason they model those separately.

When you're doing inforce modeling, typically you'll find that there have been several generations of similar products. So, it has the same basic characteristic design, but the premiums have changed over the years. The cost-of-insurance charges have changed. Typically speaking we would make these into separate model plans. We, as a rule, won't put business in different reserve bases into the same model plan. You look at your distribution of business by plan, and you decide, if you can consider some of your plans to be these small plans. I'm really only going to be modeling the major plans. So, for example, Sobel page 6, Slide 1 shows two products called Sample Life and Example Life, and there have been a couple of different generations of them. Just by looking at the distribution you can decide, well, I'm not really going to model all those actual plans. I'm only really going to model the two that are in bold in the middle, that 872 and 873, and everybody else is going to be mapped to one of them just based on materiality.

Now I will discuss modeling issue age. The most typical procedure is to take your 0- to 9-year-olds. If you want to be a tad bit more sophisticated, certainly you can look at those distributions and get your weighted average issue ages. Once in a while, if you want to get even more sophisticated, you can map on what I would call an issue age band or an issue age decade code and carry a weighted issue age throughout your mapping. Pass along the weighted issue age (IA * FaceAmt / IA * AV) in the cell mapping, and calculate the weighted average issue age for each cell at the end (or near the end) of the mapping. Obviously this is only going to work if you're going to be able to find the policy values, the data for each of those cells. Certainly on annuities, where mortality's not the same sort of risk, you'll be able to get away with fewer ages.

Sobel page 7, Slide 1 shows a simple example of what I was saying about carrying through the weighted issue age until the end of the mapping. Here I'm showing a simple example where I'm just presenting the males and the non-smokers, but we mapped all of the 0- to 9-year-olds simply to code zero. That's the MN zero in the key, and then the next one, the MN 1s, are all the 10- to 19-year-olds. And then only at the end where you look at the last two columns, and we have stored the units in the weighted issue age, in the 30-year-old bracket that comes out to a 34. Sometimes it comes out higher and sometimes it comes out lower.

Now, one thing I want to say about underwriting class is that it should reflect the actual underwriting standards that were used in writing the business. Just because you have two policies in your inforce that are labeled as P, meaning preferred, does not necessarily mean that they're going to have the same mortality if P in 1995 did not mean the same thing as preferred in 2005, and you'd probably want to separate those into separate cells.

In terms of issue year, as a fairly hard and fast rule, we do not combine or map over issue year. One of the purposes of your static validation that I will be talking about a little later is to make sure that at the beginning of your projections you have a model with an appropriate attained age. If you've already mapped across issue ages, and now you've also mapped across issue years, you're going to have too many moving parts if you have any problems with your static validation to try to fiddle with it and put it back into a better validation. Also, if you map over issue year, sometimes at the end of a maturity or at the end of a surrender charge period you'll see contracts spike. If you don't map over issue year, that will smooth some of those results out as they happen every year.

Most often in terms of issue months we model everything as being issued in the middle of the year. I've certainly seen cases and examples that are random or use a simulation approach to try to approximate the actual distribution of business within a year. Certainly if you're going to be doing short-term modeling where it's important to you to get the cash flows exactly in the right month during the upcoming year or two, you're going to want to have more cells and model things and have separate cells for issue month.

Joint life business is notoriously difficult to model and usually includes substandard ratings, which makes it even harder. Often there are a small number of policies, but those policies tend to be very large, and so typically I recommend going with a seriatim approach. One case where we did not go with a seriatim approach, where we actually did group the actual inforce, we found it useful to start by distinguishing the actual inforce. We had a last-to-die product. Which age was the youngest age? So, we grouped all the cells where the male had the youngest age separately from the ones where the female was the youngest age, and then we went through the modeling process there for issue ages on the male and then on the other life.

There are a couple of other considerations here. In terms of average size, typically we want to create separate cells by face amount bands. Instead of mapping all the business together, which usually will end up meaning that you'll really just be modeling one or two middle size bands, and you will lose any of the behavior that might be exhibited by the upper or lower bands. Similarly, on reinsurance you don't want to take some policies with no reinsurance and a handful of policies with large amounts of reinsurance, put them together and model this as some moderate amount of reinsurance.

For flexible premium products you may want to look at the premium paid to date and try to figure out some sort of average premium paid. You then want to compare that to some benchmark premiums, a minimum premium and a target premium. Perhaps the product is marketed as a 30 pay life. You can benchmark the actual average premium to those benchmarks and then try to categorize the inforce into those various buckets. When you're done, typically there are a couple of hard-to-model plans, some benefits and riders that are not going to go through the main model. What do we do in those cases? There are a couple of options. You could just do a simple one- or two-cell average model on a spreadsheet. Occasionally it's appropriate simply to gross-up your main model business and to include these hard-to-model plans. Certainly there are plenty of occasions where it's appropriate simply to ignore them.

I have some take-home points for you. The first would be flexibility. It's good to have a model that can be used for more than one purpose, and it's a good idea to ask for all the data up front so that if you have to refine your model, it's easier to step back. A general rule is to group similar business rather than averaging dissimilar business. You don't want to be averaging dissimilar business. Simply group the similar business. A friend of mine told me that this would be a bit of a downer to end on, but there are going to be problems. That's an unavoidable fact of our work, and in setting up your models it's a good idea to have in mind what you think those problems are going to be. Set up your model to help you dig in and solve them. If you think that validation is going to be a problem, perhaps you want to use a more granular model. If you think that getting good policy values, premiums and so forth, or loads is going to be a problem, you may want to go with fewer cells. If you know that the business is going to be modeled in more than one context in your company, you may want to see ahead of time if you can build a very granular model and then shrink it down for the various purposes.

MR. ROBERT W. WELANDER: I think that's really important to us when we're talking about the distinction of whether running a seriatim model or a grouped model. I think it's really hard to tell whether it's important enough to do your grouping without running both. So, oftentimes we will create both in our models, and we'll run the results side-by-side and really use that to figure out how well we did in our grouping algorithms. If you're running big stochastic jobs, in particular, it's helpful to have a grouped version of your populations. So, if you have the computing power to do so, I think it's getting more and more likely that you'll need to run them both ways.

MR. SOBEL: And when starting with a very granular model and then crunching down as need be to a more modeled version, you certainly can better quantify any sort of distortion or changes that occur from that as opposed to simply starting with that modeled version.

FROM THE FLOOR: I agree with that, and I really believe that the purpose of your model drives a lot of the decisions that you have to make. Do you have short-term needs or do you have long-term needs? You'll see this as a recurring theme, I think, through all of our discussion. There's a lot of dealer's choice here, and it is driven by your knowledge of the business, your knowledge of the model and what the needs are.

MR. WELANDER: Uri's point about the grouping on the issue ages is one mistake that I know that I've seen in our own modeling. If you always just choose the midpoint of that range as your age, so if you're using the 30 to 39 and are always saying that they're 35-year-olds, as you add more and more grouping into your models, you can get to the point where maybe that cell is only one policy anyway, and you're turning this 31-year-old into a 35-year-old. So, that's one thing to be really careful of because I've seen it quite a bit as we change our grouping algorithm.

I'm going to talk about efficient model refresh process. Before I do that, though, we're going to talk about how to model better. I think one of the relevant things we also need to talk about is why we want to model better, and to do that I'm going to tell a little story. I work for a reinsurance company. All of the data that I get from you, my clients, is old. It's never timely enough. It's never timely enough for the direct writing company. There's no way it can be timely enough for me.

Your numbers are old. I can't use them. So, how do I report reserves? I could take your numbers, and that works very well maybe in September when I have June data, but when I get into December and I have all that pre-fire sale XXX business sitting on my books, and I'm holding September reserves for it, especially when we want to hold interpolated and not mean, that doesn't do me much good. I could calculate with your data, but your data is old. That still doesn't do me any good. Plus, I have to keep my data in sync. My reserves need to stay in sync with my premiums and my claims. I can't have a disconnect. I can't show all this premium coming in and then show the offsetting reserve that grew because of it. So, how do I do it? Like I said, from you, my clients, I'm getting reserves quarterly, premiums monthly and claims weekly or maybe even daily. I have to somehow connect those.

Let's pretend it's December 20, and I get from you, my client, a closed block of data. Welander page 2, Slide 1 shows the numbers that I have as I'm preparing my December 31 financials. I have premiums through November. I have claims through part of December, and I have September reserves. Well, if I were an accountant, I'd look at the premium and average it. I'd gross-up the claim. I know it's a closed block, so my reserves are going to grade off a little over time. So, Welander page 2, Slide 2 shows what my numbers are going to look like. That doesn't seem unreasonable until the actuals show up, as seen in Welander page 3, Slide 1. The accountants didn't realize that I'm calculating using interpolated reserves, and I have a big chunk of pre-fire sale XXX business that was written in December. So, my premium is just horribly off. My claims are claims. They are what they are. Actually they did a pretty good job with that. My reserves, well, not so much. I can't afford that kind of volatility in my balance sheet. So, I have to have a way to take the data that you give me, and do something with it to make my December numbers more rational. I'm not complaining. I know that it's the best that I can get. So, you take what you can get.

Now I want to talk about an efficient model refresh process. It's really simple. How often do you model? You model as often as it's worth it to do it. There are two simple questions you have to answer before you know whether you should refresh the inputs going into your model. Do you create a better model, and is the improvement worth the effort? It's not going to be the same for different components of your model. What about populations? How often should you do it? I do it quarterly. How often should you do it? That's your call. If you have better data monthly, update it monthly if it's worth it. If it's not worth it, don't do it.

One thing to discuss is cash-flow testing. What are the concerns with cash-flow testing? If you have a really long time horizon, 30 years for example, and your purpose is to represent all of your liabilities, do you need a really granular model where the number you're concerned about is accumulated distributable earnings after 30 years? I'd argue no. Well, what about the accruals that I need to get from your September data to my 30 data? I really need granularity because at that point I'm concerned about short-term effects. Let's use a 10-year level term that switches to an annual renewable term (ART) at the end of the level period. How concerned am I about the effect of those policies? What happens to them at the end of the 10th year, if all I'm doing is accruing for three months? I'm concerned, but not too much. Now, for cash-flow testing, I'm concerned more because it has significant effects on how that business rolls. So, it all comes into play.

Is it worth it? Here's an example. Let's say that I get population data quarterly, and I happen to have a skunky lapse assumption in my model. If I don't update my population every quarter, if I let it slide once because I figure my model's good, there can be problems. If I let it slide an extra three months, and my base lapse rate is modeled at 10 percent, but it's really 6 percent, and I have a billion of reserves, all of the sudden I'm off by $10 million. That's a difference that I can't have in my balance sheet. I have to update these models. I have issues if I have bad assumptions, and especially if I don't know that they're wrong. I know my model's right. You guys know my model's right. That 10 percent must be right. The 6 percent, that's an abnormality, or the actuals are wrong. That's usually the problem.

So, let's talk about how we update those assumptions. I think this kind of falls into common sense, but every once in a while it's nice to have someone say it. The assumptions that end up in your models typically start with pricing, and if you're a pricing actuary looking for how to develop these assumptions and you don't have pricing available to you, look for something similar. Look for products written in the same era, look for something written recently that mimics it, and use those numbers to start to develop something that looks like the business you're trying to model. And obviously, as experience develops, take advantage of it.

Use actuarial expertise. That's like the catchall for everything. You have maybe one data point that you're telling a story around. Your gut tells you you're right. So, it's actuarial expertise. You'll hear me use that some. Welander page 21, Slide 2 takes a look at an example where actuarial expertise comes into play. You did a mortality study. You used your pricing basis as an expectation. You ran it through, and your actual to expected (a-to-e) is 105 percent. Well, then updating your mortality's simple, isn't it? You just put an aggregate 1.05 multiplier across the top and you're done. Maybe not. Not all business acts similarly. Sometimes it's sold differently. Sometimes there are different products. So, let's take a look at that.

Welander page 22, Slide 1 breaks this out by channel. I called it channel because for me, as a reinsurer, channel is ceding company. For you, channel could be distribution system, for example. You have brokers. You have direct writing agents. However you break out your business, there's your a-to-e ratio. Well, all of the sudden it's not just 1.05 anymore. You have to look at it and understand your business. One of the keys to everything that we're going to talk about is understanding your underlying business, and it's not just knowing how to calculate a Commissioners' Reserve Valuation Method (CRVM) reserve. It's understanding how it was sold. It's understanding when it was sold. It's understanding the underwriting around when it was sold. When I look at these distribution channels, I start to think I may have had underwriting issues in some of those channels.

So, now that this is in place I can say, oh, well, see, I'm smarter than just that 1.05. I know it's different by distribution channel. When I put assumptions in my projects I can just load these in by distribution channel, and I'm good to go, right? Maybe not so much, because a smart actuary knows that it's not always just distribution channel. There can be other factors that drive differences in your experience.

Welander page 22, Slide 2 takes a look at another one. It's not just distribution channel. Now my products have different experience, and I think we've all seen this within our own products. We're seeing that we underwrote them the same way, but either the agents or the policyholders anti-selected against this. Okay, that's part of the game. You have to recognize it when you build your models, otherwise as you project farther out than three months, you're going to start putting numbers in front of management that aren't really going to prove to be anywhere close to correct. So, now you're faced with a conundrum. How do I model my mortality? Do I do it by product? Do I do it by distribution channel? Do I just put 5 percent across the top?

Take a look at Welander page 23, Slide 1. I split it out, and that's how the numbers fall together when you combine the two views. Well, then this is the right answer, right? Maybe. It depends on how well you know your business. And the question you have to keep asking yourself through every step of this analysis is at what point do I lose credibility in the underlying number? Is my mortality that Channel 5 Product C; that 104 percent? That may be three policies. Three deaths tends not to give you too much credibility. It comes down to how well you know your business and how you make that decision.

Since we're pretending that every single one of these numbers is credible, you decide that this is the level at which it's appropriate for you to modify the mortality assumptions in your liability model. So you do it. You load them in. Good for you. You're done, right? Okay, here's the reality. You know that you put it in right and all the work you did was right. Does your boss? Does your chief financial officer (CFO)? Does whoever's going to get yelled at when it's wrong know it's right? No? So, what do you do about it? First of all, communicate.

This is an example of where you have to communicate these numbers. You can't be the only one who knows it and believes it. That just leads to issues later. Not only do you need to communicate it, you need to show the cause and effect. You can't just say, oh, I loaded these numbers, now I'm going to lunch. You have to run the model. You have to rerun it. You have to say using our old assumptions here's what the result here looks like. Was it your cash-flow testing number? Was it your reserve at the end of three months? Was it distributable earnings over the next three years? Here's what it used to be. Here's what it is now. Here's what changed. Here's why it changed. It's all about communication.

Sarbanes-Oxley (SOX) is everywhere, and this isn't an SOX issue unless these affect your GAAP numbers. You're going to have to get signoff. You're going to have to get someone to look at it and say, yep, you did it right. You also have to document. I can't say it enough. If you don't document it, it doesn't count. If you don't document it, no one knows why those numbers change. You may know why, but six months from now are you going to remember why? Maybe or maybe not. Six months from now when you've taken another job, is the person you trained in a week and a half to do this going to know why? Are they going to know how you even came up with it? Document it.

MR. SOBEL: The one thought I was having when you were talking about it being worth it is that very often, at least in our work, you really don't know whether or not it was worth it until you've done it, and that happens all the time where you need to refresh your model. You honestly don't know the impact it's going to have, and, unfortunately you finish going through the modeling process, you run it all through and get answers, which are materially very much the same as you had before. Doing this whole refresh hasn't added any particularly new information except that it's maintained itself. That's a case that we get into a lot, and to some extent I find a lot of automation helps with that issue. If you're going to have to be updating on a regular basis, and you don't know if it's going to come out the same or not, you want to get the automation in place for that model building process.

MR. WELANDER: I agree, and I think automation is something that we all get to look at and decide if that comes into the equation of: Is it worth it? It isn’t just around do my numbers get better, but how hard was it for me to do that? Is it really, really simple? Who cares if the benefit is incremental? Do it anyway because it was really, really simple.

MR. SOBEL: Right. And often you won't even know until you've done it whether or not it was worth it.

MR. WELANDER: I agree. However, I think sometimes just going through the process itself can make it worthwhile. I remember we were in a situation where we were updating our models every year, just once a year. It seemed like every time we did that it was so long from when we did it last time that you'd forget how everything worked, and the process wasn't fresh in your mind. So, moving to more of a quarterly or a monthly process, even if it's not adding value and changing things dramatically each period, can really help you kind of speed up and catch errors and make improvements.

MR. SOBEL: Well, that's a good point also. If you've moved from an annual model refresh process to a quarterly, you very well may see fewer changes in each successive quarter whereas before you were seeing larger changes over that yearly refresh frequency.

MR. JEFFERY A. FITCH: I'm going to talk about model organization and validation. First of all, to kind of put my presentation in context, I work for Principal Financial Group, which is a large multi-line financial services company. I work in our corporate area, and my main responsibility is organizing and coordinating our overall company modeling activities and working on and enhancing our modeling capabilities throughout the company, as well as trying to roll out various business unit financial projection models up to a total company basis. My main focus is really on GAAP projections, enterprise risk management and cash-flow testing.

As our models are getting more and more complex, it's becoming exceedingly important to make sure that they're well organized. Here are a few steps to consider as you're building a model to make sure that it is well organized. First of all, before you do anything, figure out what you are going to use this model for both now and potentially in the future. Is it just a pricing model that's going to be used for this particular product? Is it something that's going to be used and expanded upon and rolled up to total company models? Those types of questions could help you kind of determine how you might want to build things.

Secondly, make sure that you define the resources that you need for your model. If you're like me, you're very reliant upon various other sources, either other experts within your company, data that feeds into your model extracts or assumptions. Make sure that that's all defined up front. Make sure that you're setting up a process for documentation as well as control standards up front, and I'm going to go into a little more detail about control standards later.

The next step is validation processes. I'm going to spend a lot of time talking about model validation, what some of those processes are and what they need to be, but that's something that you really need to think about up front. You should figure out what type of output you need from your model and what the end result is going to be. What are you trying to capture from your model? Make sure that you're building things such that you can capture that level of information. Finally, after doing all that, now you can really start building the model and documenting it and making sure that that process is done simultaneously. I've seen many times that you'll go through a whole model-building exercise, and then at the very end you'll try to pull together your documentation, but a lot of times you'll forget how things were done, and it really needs to be a simultaneous effort. So, the key here is really to plan ahead, and too often I think we just jump right in and start building our models because we get so excited to do that.

Now I want to discuss model uses. Traditionally there was more of a distinction between pricing or new business type models and maybe an inforce model used for cash-flow testing. I think now we're seeing a lot more blurs between those traditional lines, and models are used for various applications. They include cash-flow testing; full-blown GAAP financial projections; asset/liability management; risk analysis, which seems to be an expanding modeling need as you move to more of a ERM framework; valuation, which might be reserve calculations on different valuation bases, as well as valuing certain blocks of business for mergers and acquisitions; and economic value added or embedded value calculation. With all these different potential uses for modeling, you might ask, Can one model do all these things or do you need a separate model geared towards each one of these different applications? So, let's talk about that in a little more detail.

There are some advantages of having a common model throughout your company or organization that can be used to do various things. First of all, I think it provides a link between different functional areas. Particular for us is a link between the pricing and the financial reporting or projection area. If you have the same model used for those different sources or different applications, it really helps to kind of tie those two areas together.

Also important is that it leads to improvements to all models; helps develop a consistent set of assumptions and calculations; allows new products and features to be added only once; and develops efficiencies.

It allows more time for analysis by avoiding building duplicate models that are used for the same or similar purposes. So we want to have the same model, the same underlying calculations and the same assumptions, but potentially different levels of aggregation for different purposes. As Uri talked about, you might have a pricing model or a business unit short-term projection model where you decide it's very important for us to group this business monthly or seriatim, whereas maybe in your cash-flow testing or your longer term projections or total company models that's not nearly as important. So, having the ability to expand and collapse the granularity of your model is important.

I think it's important as you have a full-blown asset/liability model to be able to easily take assets out of the equation hopefully by making some sort of assumption as far as what your portfolio rate might be. I think getting past that point really helps some actuaries that are focused more on the liabilities get past that complexity, and a lot of times if they see this huge model with all this asset stuff in there they can get confused. Now, don't get me wrong. I think assets are a very important part of the modeling and balance sheet. You need to get that done right, but I think having the ability to turn that off or on is also important. We felt that it's important to have one person for any given business unit that crosses these different applications. That one person is used to make sure that they're the ones doing the main maintenance of the model and making the code changes. That person's going to rely on various experts from within the company.

Now I want to talk about control standards. You need to lay out who can make changes to the model and even maybe when those changes can occur. It's an important part of model organization. A lot of times today with the flexibility that you have in a lot of the modeling software out there, it can be a curse if you have everybody in there making changes to the system, to their various products. If you don't have a well-organized effort of how people can make changes and how it's going to be documented, it can get out of control pretty quickly. We've developed internal standards of practice for how to make changes. They cover even simple things, such as when you do make a change to the underlying code how to put comment in, and to put your name and date and a reason for the code change. It basically gives you a consistent way to keep track of any modifications that you make to your model.

Peer review is also an important part of this. Having a regular process for other people reviewing your models and practices is an important part of control standards. It seems like we're all under a lot more scrutiny these days with Sarbanes-Oxley and really need to make sure that we have our documentation and peer review in order. Version control is another important aspect of model organization and control standards, and that means as you get different versions of your underlying software, as upgrades are made, you need to make sure to migrate your custom changes to the new system. Setting up a process that's clear on how you're going to move forward with that is important.

Now I'm going to touch base a little bit on model validation. This is an extremely important exercise for having a credible model, but I can't underestimate the amount of time the model validation sometimes takes. It seems like it's almost a never-ending process, and if you're like me, it seems like you're never truly happy with how well your model validates, and so it's a process that you can continue to improve on. A lot of times it's hard to put that down, and maybe you're not as close to what you like, but because of time reasons you just need to continue to make improvements. One thing that's helped us with this is to set up a checklist for validation. In advance, as part of your initial model organization, kind of check out those items that you want to validate, whether they're particular line items or whatever, and as you're going through make sure to actually physically check those things off and document and say how close you were to various things.

Again, it is an iterative process, and I think it does link into both Uri and Rob's presentation on, for example, assumption setting. You might find as you're going through and validating your model that a particular line might not make sense, and maybe it's a problem with the underlying assumption. Maybe that drives you to do an experience study, or maybe it ties into your model point selection. If you're not as close as you might like, maybe it's because of how you group some of the underlying problems. So, again, it's an iterative and continuous process and not something that you can say, okay, my model's validated.

I'd break validation into both static and dynamic validation, and I'll kind of talk through what that means. With static validation, what you're really looking at is the starting balances of your model as of the valuation date or starting point. The focus here is on balance sheet and statistical type items, things like reserves under the various bases, policy information, account value and benefits. That might be your life insurance benefit. It might be a disability benefit, annuity benefits, asset information or GAAP totals. One thing that I kind of throw into the static validation process that sometimes isn't thought about is what your initial asset portfolio rate is coming out of your model. You should make sure that really does match with what your underlying assets are earning, as well as your initial credited rate that's coming out of your model, and sometimes the difference between the two, that portfolio rate less the crediting rate, because that's a big driver of profitability. You should make sure that those do match known sources. As you're doing the static validation or any validation, it's important to define what your tolerances are up front and how close you want to be on these different items. You should document that and really try to improve that model fit over time.

As I mentioned before, the validation process really can tie into your model point selection. You might be comparing reserves, for example, on a block of immediate annuities, and you're just not close enough. So maybe it was a problem with how you were grouping those underlying liabilities, and I think again, it's an iterative process where you have to go back and forth. One thing to be careful of is to avoid offsetting errors when you're doing validation and make sure that you're validating at the lowest level possible, maybe by plan code, or even by policy or groups of policies. A lot of times if you set thresholds ahead of time, like I want to do my validation until I'm at least 5 percent of the actual value, once you get to that point you stop your work, but there might be a lot of underlying offsetting errors that you should dig into further.

I often break the static validation process into two steps. The first step is validating the underlying extract files themselves. That is the information that's maybe feeding from your administration system or whatever source that you're using to feed into your model. You need to be making sure that those match other known sources, such as actual financials or other reports. The next step, now you have all this information in your model, is to compare or validate the initial model balances to those other sources. That's very important in particular for calculated items in your model. A good example might be a reserve calculation, where you are comparing your calculated model reserves to an actual financial result.

Now we've done our static validation which is really one point in time. Moving forward to dynamic validation, the focus here is on your projected values in your model, both income statement, cash-flow-type items, as well as projected future balance sheet and statistical type items. Again, with this there are just so many different things that you can look at. I think that’s why it's called dynamic validation. One of the things you really want to make sure of is that the dynamics of your model are working correctly or as you would expect, making sure that things are behaving correctly under different economic scenarios, for example. You might have a dynamic lapse formula for a block of single premium deferred annuity (SPDA) business. Checking that and modeling that under different interest rate scenarios is an important exercise to do. Again, as with any validation, make sure that you're not just looking at the bottom line result, but that you're looking at intermediate values. For example, if you're trying to validate your projected earnings, you might have earnings, but it might be due to offsetting errors that you have on both the revenue and expenses side of your income statement.

When doing dynamic validation, one way you can do that is as you're going through your model refresh or model upgrade process is to compare your projected results from one model to your previous version of that model. That's an important thing to know. As you're going forward, did results change and why? Make sure that you document and understand that, and that can be a good way to even uncover any problems with your model refresh. In the cases where you might have multiple models, you can validate your model to other projection systems or expectations that you have.

Another important means of validation is to validate your model to actual financial results. An example that I'll use here is let's say that you refresh your model with a valuation date in the past, December 31, 2004, for example. Then populate in your model the key drivers that you have. Look at your actual sales and put that in as your new business liability sales assumption within your model, as well as kind of populate those main economic drivers. They might be interest rate, equity returns, etc. Once you've done that, then you really try to compare. Given this information that I know, how well did my model do at predicting actual results? And that can be a time-consuming process. There can be a lot of things that cause results to vary from what's in your model, but I think it's an important exercise. Really start by looking at those results line-by-line. As you find errors, and you will, you'll find differences, and they might not necessarily be errors that can cause you to look more closely at other assumptions.

Let's say you're looking at projected surrender benefits in your model, and it's way off from what the actual result was. Well, is that a problem with your underlying assumption or was there maybe a period where surrenders for one reason or another were much higher or lower than what was expected? Was it a random fluctuation or was it really a faulty assumption that you have? Or maybe the dynamics of your model weren't working as you hoped or expected. Maybe it was a period of low interest rates, and your model was assuming that these policies react much faster than what they actually did to the change in economic assumptions. So, there are a lot of things that you can look at, and I think, again, it's an iterative process that causes you to go back and forth and really have to use that actuarial expertise. Is this really a problem with our underlying assumption or is it just a random fluctuation?

I'm going to wrap this up by talking about ways to remove the black box nature of models. As our different models get more and more complex, one of the potential problems is that you become so reliant upon the model results that they become a black box. Models are important tools, as we all know, but they can't replace things such as business sense, judgment, reasonableness, etc. There are a lot of things these days that are leading to more of a black box nature in some of these calculations. There are a lot of complex reserve calculations, option pricing techniques, stochastic modeling and things that are very difficult to get your hands around. So, I'll talk about a few ways to make sure that you're not causing problems in your modeling and that you can basically explain the results. I think that's what the bottom line is here. I've been in situations where I’m explaining results to management and my model spits out a particular result that maybe I can't explain, and my answer is, well, that's what the model says. That's really not an acceptable answer, I don't think, and there are ways to get beyond that. Basically, I think just checking for reasonableness and getting out your calculator helps. What do you expect the results to be for particular line items or scenarios? Really try to develop that business sense as opposed to just model reliance itself.

One thing that we've done for a lot of the more complex calculations is to make sure that you can take a piece of your model and reproduce those model calculations within Excel. For example, you can take a reserve calculation for a given policy and exactly try to replicate what the model's doing or take an account value roll-forward and make sure that you really do understand the pieces of that. Now, you don't want to get too carried away with this and basically replicate your entire model within Excel, within another package. That kind of defeats the purpose, but I think it can be an important training tool as you have new people moving into your particular area or are trying to teach people how the modeling software works. Having simple tools within Excel or other platforms will show them how things are working.

Another important way of removing the black box nature of models is to make sure that you can get detailed model output. Make sure you have the ability to produce things such as audit reports for complex calculations. Make sure that you have the ability to drill down into the details of your underlying model calculations, like getting to the results by cell or by groups of policies. Having access to that level of detail is important. There can be some technical challenges with that as well. As your models get huge, if you're really producing data or information at that granular level of detail, it can get a little bit overwhelming. The key is to just start simple, making sure you're adding one feature at a time and testing the results of that. When you're doing modeling you'll start with one scenario, maybe your level interest rate scenario, then start looking at a few different deterministic scenarios. Don't jump right in and do your full-blown stochastic analysis until you really understand the pieces of that.

MR. WELANDER: You have to have one person who understands the changes that have gone into this and ultimately should have ownership for the effects. I think that's critical. Also, you were talking about validation. We've actually combined validation with communication and gotten to the point where if we implement new models into our process, we make whoever puts those models in place report out to management and put a PowerPoint presentation in place to explain exactly what they did, why they did it and what the effects were. That validation is critical at the front end because it gives credibility. If you get it right the first time, all you have to do is monitor it, not question at every corner if you looked at that block properly.

MR. SOBEL: You don't always have the luxury, but we have several projects where when we're putting in different assumption changes or coding changes, step-by-step every change gets quantified, either in a cash amount or in a return kind of calculation. We have found that to be extraordinarily helpful on many levels. One is it helps you get the business sense on what's driving these changes rather than doing it afterwards. It also really helps with the documentation process when each piece is step-by-step quantified as it's changed in your model. Another comment struck me while you were talking about looking at your static validations at a refined level, not just in aggregate. You even said something about looking sometimes on a cell-by-cell, policy-by-policy basis, and it's been my experience almost without exception looking at the results of a static validation on a cell-by-cell basis, when you take a look at those far outliers where things are validating at 1 percent or 2,079 percent, that always reveals something that you've forgotten. That's actually driving what's going on inside the model. Even with some of the normal cells there may be some considerations you've missed, and I highly recommend looking at those validations from worst to best and taking a look at those ends. It's always helped us.

FROM THE FLOOR: That's a good point. And as you dig into some of those ends or outliers, it really does tend to fix or correct other things within your model as well. So, if you start at the ends, that's really going to help improve your overall model fit. You might have a threshold in which in total you want these modeled results to be within, say, 1 percent of actual results, but your threshold for error needs to be wider when you're looking at either individual cells or groups of cells. Otherwise you're going to drive yourself crazy doing continuous validation.

MR. SOBEL: Typically speaking, are you able to put into the models themselves the sort of documentation that you want, as opposed to in separate documents that are supposed to follow along with the model but don't always make their way? I've always found that it's helpful but difficult if you're using a software package, to put appropriate documentation all the way into the system itself because that will pretty much always follow along, and it won't get dropped. The systems aren't set up in some ways to facilitate that.

MR. WELANDER: I'd agree with that completely. We tend to document within the model two different ways. Although I can’t say we require it, when we're actually modifying the underlying code, we strong-arm people into putting documentation around every piece of code that they upgrade and putting their initials and putting a date so that we know who did it and who to blame when it's wrong and how old it was. So, there may actually be a real reason for it to be wrong, but then when you get into actually coding the assumptions that becomes tricky, and the only approach that we found that has any usefulness to us is in the table naming convention.

Here's kind of a stupid example, but when we're coding in mortality assumptions within a particular plan code within our model we will code it. We won't code it in as a number. We'll code it in as a named table that's named 60 percent. If we coded it as a reference number, when we look at that field, if I wanted to know what that number was, I had to dig a step further. Now I can look at the table name and say, oh, my mortality assumption is 60 percent. Typically, the way those numbers get coded is you get three male entries and three female entries, for example, just one right on top of the other. So, instead of having to dig in six times to see what those numbers are, you can look at the table name and see that it went from 110 percent to 72 percent to 27 percent and on down the list. That's as close as we get to documenting within the model.

MR. FITCH: I think both levels of documentation are extremely important. You can't rely totally upon documentation within your model because that's hard to share with others. It's hard to share with management or someone that doesn't have an understanding of the underlying software. So, I think having a separate stand-alone Word document that really does detail that out is extremely important, but for the people that are digging in the models it's nice to know as you add new variables or new calculations that it's clear within the model itself what they do.

MR. WELANDER: To take it a step further, at the high level, once you've implemented whatever changes you're going to make, it almost becomes a corporate cultural issue as to how well that gets documented. There are some corporate cultures that require you to present it out to make sure management understands it, and that is self-documented. You have a pitch in place. You have an e-mail in place. You have minutes from a meeting in place that talk through exactly what you did, why you did this, and management saying, okay, I agree, except you have to answer these questions for me first. There are some places where you do it, you tell your boss, and he says okay, and you're done. It's really a corporate thing. Some of it's personal. Some people are really good documenters. Some people are at the other end of that spectrum, and some of it's personal, but some of it's the company culture.

FROM THE FLOOR: It is pretty astounding how much time you could spend on the documentation at the different levels. Those e-mails are good in some cases, but, of course, at the end you need some more unified document that shows everything more in one place to avoid that shopping cart issue. If you have a separate report with just the assumptions, let's say, each time you refresh the model, hopefully the structure of that report stays relatively the same and it's not as difficult to update each time.

MR. JOHN WEUM: Jeff, in regards to use of common models and the compromise there, what's been your experience? Do the common model and the assumptions in it still drive your level of aggregation? How you aggregate? Do you have flexibility to adjust assumptions to fit the aggregation level?

MR. FITCH: We try to make sure when we're developing assumptions like, for example, mortality assumptions, when we're building a table within our model, that we make sure that we build those things such that they could work for whatever level of aggregation we end up deciding to use. You don't want to go in and build in ages for every 10th year, and then try to model something on a seriatim basis. You're not capturing that. So, that assumption development up front needs to be done such that it captures the finest level of aggregation that you can potentially handle.

MR. TOM HAMM: Uri, we develop the granularity method similar to what you described using a relational database in looking at the distributions. I was wondering if anyone had any experience using one of the commercial model builders that often come with your actuarial software.

MR. SOBEL: Our model builders tend to come with first and last names. We have people who typically have their systems that have been set up for years. They're the experts at it. And most of it is done either in Access or in Excel. There are some people who have had luck with some of the commercial versions. I think to some extent my experience has been that the learning curve on Access or Excel is a little shorter and gets you pretty much to the same place. The one piece which I find the commercial packages to be very good for on the static validation side is if you can feed in your actual into your model so that when the model is actually done you have no concerns that you're comparing apples to apples. We used to be in a situation where it wasn't possible to do that feeding in, and we'd create a model outside in Excel or Access. We'd have all of our policy values outside, and we could even do our validations before ever running the model, and that was very helpful. There's a certain advantage to that. On the other hand, when you're done you always have to go back and check to see if your validation on the outside matches what's actually being produced in the industry software. So, I would say that the industry software is helpful in terms of the validation stuff. I've only used Excel and Access for the model building process.

FROM THE FLOOR: Yes, that's a good point with bringing in particular items that you're calculating within your model, like reserves, for example. You might not need that information as far as what the actual reserves on a policy-by-policy basis, but I think it's important to bring that into your model itself so that you can do that side-by-side comparison at whatever level of aggregation you end up deciding.

MR. FITCH: And I would say that feeds into my prior comment about asking for every field you can think of. I had in mind, in particular, the validating process. So, you ask for actual reserves. You ask for actual premiums inforce. You ask for anything you can think of. If you think you might need it, ask for it.

MR. SOBEL: I guess I'm the third strike there. We built our own also, and a lot of the logic was so that we could customize the reporting that we got out of it. We could look at the statistics that came out of it the way we wanted to see it, not the way the commercial company decided that I wanted to see it.

MR. WELANDER: Something I thought as Jeff was talking was the communication between the pricing. Once it gets out of pricing's hand, it goes somewhere, such as valuation or some corporate area, wherever it is. It's critical to make sure that pricing views the world the same way that the valuation people do. Pricing may be using mean reserves and pricing over the level period only on a level term with an ART at the end. Valuation looks at it completely differently and may have enough credible experience out into the tail to say we can't afford just to price during the level period. If we do that, we're going to price ourselves out of the market, or whatever the case is, whether it's a plus or minus. So, with that communication, and understanding those rules, I think there has to be consistency.

MR. FITCH: I agree. I think it goes both ways as well. I think by having that tie between your pricing and your financial reporting it really helps you understand the impact of pricing decisions. But also, as you're making improvements or as you're validating things and realizing that maybe those pricing assumptions didn't hold true, by having a common model it allows instant feedback to the pricing unit. You will be able to say, well, I thought I was at this internal rate of return (IRR) or whatever, but based upon this distribution of business or this assumption, which we now feel more comfortable about, we're really here. So, that really helps that tie as well.